Online Dynamic Value System for Machine Learning

Haibo He[1] and Janusz A. Starzyk[2]

[1] Deptartment of Electrical and Computer Engineering, Stevens Institute of Technology, Hoboken, NJ 07030, USA. <hhe@stevens.edu>
[2] School of Electrical Engineering and Computer Science, Ohio University, Athens, OH 45701, USA. <starzyk@bobcat.ent.ohiou.edu>

Abstract. A novel online dynamic value system for machine learning is proposed in this paper. The proposed system has a dual network structure: data processing network (DPN) and information evaluation network (IEN). The DPN is responsible for numerical data processing, including input space transformation and online dynamic data fitting. The IEN evaluates results provided by DPN. A dynamic three-curve fitting (TCF) scheme provides statistical bounds to the curve fitting according to data distribution. The system uses a shift register communication channel. Application of the proposed value system to the financial analysis (bank prime loan rate prediction) is used to illustrate the effectiveness of the proposed system.

Key words: Value System, Machine Learning, Reinforcement Learning, Curve Fitting, Financial Prediction

1 Introduction

Online value system is useful for machine learning. For instance, in reinforcement learning (RL) a machine learns values of its state/action pairs [1] to direct its actions towards a goal. By analyzing sensory inputs from the external environment, an intelligent system (agent) should evaluate the information received according to its value system, and act to maximize the expected reward. An agent learns from active interaction with its environment, and while acting on the environment, it accumulates knowledge through experience.

A typical reinforcement learning system includes the external environment, a policy, and a value function that describes expected reward. R. S. Sutton argued that in this system the value function is of critical importance as all RL algorithms estimate the state-action values [1]. Although it is important to estimate the value accurately and dynamically, it is difficult to do so in practical learning environment for numerous reasons:

- Limited availability of information;
- Information ambiguity and redundancy;
- High dimensionality of the data set;

– Time variability of the information;

Due to the importance of the value systems, many research results have been recently reported in the literature. For instance, paper [2] proposed an artificial neural network value system incorporating multiple regression analysis. This system combined the analysis results of three neural networks, named back-propagation, probabilistic network and self-organizing feature map. Paper [3] proposed a fuzzy-based navigation system for two mobile robots using distributed value function reinforcement learning. This approach enables multiple robots to learn a value function, which is an estimation of future rewards for all robots. In this way, cooperations between two robots are maintained and each robot learns to execute the actions that are good for the team.

In this paper, we propose a novel online feedforward neural network value system capable of estimating the value of multi-dimensional data sets. Similar to learning array presented in [4], this network has a multilayer, regular array structure of processing elements (PE) with local interconnections that can be determined by PE self-organization scheme.

The motivation of this research is to provide a mechanism for the intelligent machines to be able to dynamically estimate the value function in reinforcement learning (specify "good" from "bad"), therefore guiding the machines to adjust its actions to achieve the goal. The "value" in this paper can be a numerical expression of a fundamental principle or desired objective function value in a practical application problem. A user can define his own value for each application. For example, in financial analysis, the value could be a numerical index that reflects the intrinsic value of the analyzed company for an investment decision or a numerical measure of its financial performance. This approach differs from classical backpropagation neural network approach in which a function value is given as a desired output and is used to adjust interconnection weights in the backpropagation process [5].

The main contribution of our research is the proposed dynamic value system and its implementation architecture. This value system is a scheme, not a specific algorithm; therefore, it can be used in different ways, such as selection of input space transform functions, selection of different basis functions or different voting schemes.

2 Online Curve Fitting Principles

Online dynamic curve fitting is the core module of the proposed value system. It contains a network of processing elements (PE) that approximate the incoming data values. In this section, we first show how PEs implement online dynamic curve fitting. We then discuss the proposed three curve fitting (TCF) scheme to fit the statistically distributed incoming data.

2.1 Online dynamical curve fitting

Consider a dynamic adjustment of the fit function described by a linear combination of basis functions φ_i,$i = 1, 2, ...q$, where q is the number of basis functions.

This number can be adjusted according to the required accuracy and the data noise level. Our objective is to dynamically fit values of the received data samples. We assume that each PE has two inputs describing its subspace data points with coordinates x and y. Each PE dynamically adjusts its fit function to minimize the least square error in approximated values of all the training data x and y as follows:

$$Y = a_1 \times \varphi_1 + a_2 \times \varphi_2 + ... + a_q \times \varphi_q \tag{1}$$

We can determine the coefficients $a_1, a_2, ...a_q$ by pseudo inverse of the matrix composed of the basis function values at the input data. To do this dynamically we need to accumulate function values and their combinations for different input samples. Thus the unknown coefficients a_i in equation (1) can be solved as follows:

$$Y = \begin{bmatrix} \varphi_1 \ \varphi_2 \ ... \ \varphi_q \end{bmatrix} = \begin{bmatrix} a_1 \\ a_2 \\ ... \\ a_q \end{bmatrix} = \Phi \times A \tag{2}$$

then we have

$$\begin{bmatrix} a_1 \\ a_2 \\ ... \\ a_q \end{bmatrix} = \left(\Phi^T \Phi\right)^{-1} \Phi^T Y = \begin{bmatrix} \sum_{i=1}^{n} \Phi_{1i}\Phi_{1i} & \sum_{i=1}^{n} \Phi_{1i}\Phi_{2i} & ... & \sum_{i=1}^{n} \Phi_{1i}\Phi_{qi} \\ \sum_{i=1}^{n} \Phi_{1i}\Phi_{2i} & \sum_{i=1}^{n} \Phi_{2i}\Phi_{2i} & ... & \sum_{i=1}^{n} \Phi_{2i}\Phi_{qi} \\ ... & & & \\ \sum_{i=1}^{n} \Phi_{1i}\Phi_{qi} & \sum_{i=1}^{n} \Phi_{2i}\Phi_{qi} & ... & \sum_{i=1}^{n} \Phi_{qi}\Phi_{qi} \end{bmatrix}^{-1} * \begin{bmatrix} \sum_{i=1}^{n} \Phi_{1i} Y_i \\ \sum_{i=1}^{n} \Phi_{2i} Y_i \\ ... \\ \sum_{i=1}^{n} \Phi_{qi} Y_i \end{bmatrix} \tag{3}$$

where n is the number of data points. For online implementation, this requires storage of $q \times (q+1)/2 + q$ values of different correlations in equation (4)

$$Y = \begin{cases} \sum_{i=1}^{n} \Phi_{ki}\Phi_{mi} \\ \\ \sum_{i=1}^{n} \Phi_{ki} Y_i \end{cases} \tag{4}$$

where $k, m = 1, 2...q$. As new samples arrive, these s values are updated, and equation (3) is solved for new coefficients $a_1, a_2, ...a_q$. In general, for q basis functions we may need to invert $q \times q$ matrix $(\Phi^T \Phi)$ to update the coefficients of the approximating equation.

2.2 Three-curve fitting and the voting scheme

For noisy data, the single curve fitting technique presented in section 2.1 has its limitations. Fig. 1(a) gives a general idea of such a single curve fit by an individual

PE. As we can see, the fitted curve does not reflect the statistical distribution of the input data values in areas A and B, which will cause poor value fitting in these areas. We could compute a standard deviation of the approximated data from the curve fit, but this would only give a uniform measure of statistical errors that does not reflect the different quality of approximation in different regions of the input space. In order to overcome this limitation, a three-curve-fitting (TCF) scheme is proposed. Fig. 1(b) illustrates how the TCF method, fits the same sample data values by using three curves:

Neutral Curve: that fits to all the data samples in the input space (same as the curve in Fig. 1(a))

Upper Curve: that fits only to the data points that are above the neutral curve.

Lower Curve: that fits only to the data points that are below the neutral curve.

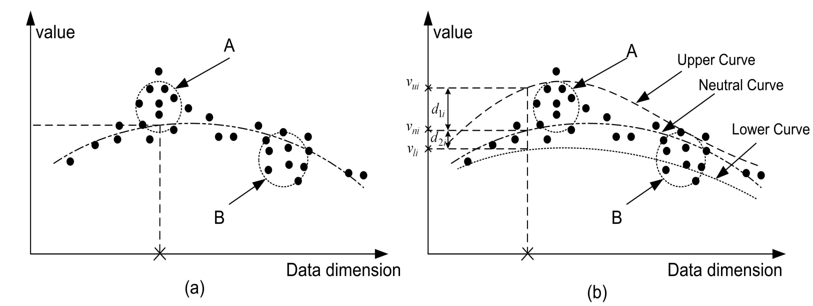

Fig. 1. (a) Single curve fitting; (b) Three curve fitting (TCF) scheme

As we can see from Fig. 1(b), the *neutral curve* provides a rough estimation of the fitted value, while the *upper* and the *lower* curves provide localized statistical distribution information. In a dynamic implementation, when a new data sample is received, we first modify the *neutral curve*. Then we calculate the fitted value of the *neutral curve* v_{ni}. If v_{ni} is smaller than the true value of this new sample, then we continue to modify the coefficients of the *upper curve* and keep the *lower curve* unchanged; otherwise, we modify the *lower curve* and keep the *upper curve* unchanged. Based on these *upper* and *lower* curves, we can locally characterize a statistical deviation of the approximated data from the value estimated by the *neutral curve*. As illustrated in Fig. 1(b), v_{ui}, v_{ni} and v_{li} are the values estimated by the *upper curve*, *neutral curve* and *lower curve*, respectively. The standard deviation of the estimated value is defined in the following way:

$$d_{1i} = |v_{ni} - v_{ui}|\,, d_{2i} = |v_{ni} - v_{li}|\,, d_i = (d_{1i} + d_{2i})/2 \tag{5}$$

d_i reflects how accurate the estimated value v_{ni} is compared to its true value. Small values of d_i mean that v_{ni} is obtained with higher confidence and should carry higher weight in the voting scheme at the information evaluation network. However, when d_i is large, it means that v_{ni} is not so accurate and should contribute less to the final result. Therefore, the weights for each PE are calculated as $w_i = 1/d_i$. For a value system with k processing elements, the voting mechanisms used in the IEN network is implemented through:

$$v_{vote} = \sum_{i=1}^{k}(v_{ni}w_i) / \sum_{i=1}^{k}(w_i) \tag{6}$$

The average weight of all inputs processed by a PE can be used as a measure of quality of the local fit to the function approximated by this PE. This in turn, can be used by the PE to select a subset of inputs from all inputs connected to this PE, and to perform the dynamic function approximation in the subspace based on the selected inputs. Each PE selects its inputs such that its average weight is maximized. It can do it locally, independent on the state and the interconnection scheme of other PEs. This results in topological self-organization similar to the one presented in [4].

3 Value system architecture

Fig. 2 shows the architecture of the proposed value system with dual network structure composed of DPN and IEN. DPN contains multiple layers of data processing elements (DPE). Each DPE selects its inputs, conducts the three-curve fitting as discussed in Section 2, and outputs their fitted values v_{ni}, v_{ui} and v_{li} to be processed by voting processing element (VPE) in IEN. The VPE establishes the final value using equations (5)-(6).

This architecture channels the information in a way similar to a hybrid shift-register structure. Each DPE has a set of inputs, that are pseudo randomly connected to local input channels. At the first clock cycle, input data is available at the first layer channel, and the first layer DPEs read this data as their inputs. DPEs output the transformed data (using their local transformation functions) into the same locations at the input channel. They also output their estimated values v_{ni} and their corresponding weights w_i to the VPE in IEN network. IEN network is composed of the sequence of VPE elements terminated with an element that computes the final value. The VPE combines values and weights received from a single layer of DPEs according to the following equations and passes them to the next layer of VPEs at the next clock cycle.

$$\hat{v}_l = \sum(w_i v_i)_l + \hat{v}_{l-1} \tag{7}$$

$$\hat{w}_l = \sum(w_i)_l + \hat{w}_{l-1} \tag{8}$$

here the subscript "l " means the values obtained from the layer l . Therefore, $\hat{v}_l$ and $\hat{w}_l$ represent the combined value and weight information for layer l . Final values are estimated by computing the ratio of the last layer $\hat{v}_l$ and $\hat{w}_l$.

At the next clock cycle, the transformed data (the output data of the DPE in the first layer) is shifted as the input data to the DPEs in the second layer, while another set of input data samples are sent to the first layer channel. The VPEs in the second layer combine the results obtained in the second layer with that passed from the previous layer. Other layers process their corresponding information concurrently, implementing a hybrid pipeline structure for function vale estimation. In this way, all processing elements in the system are active during all clock cycles making this architecture suitable for the dynamic online processing.

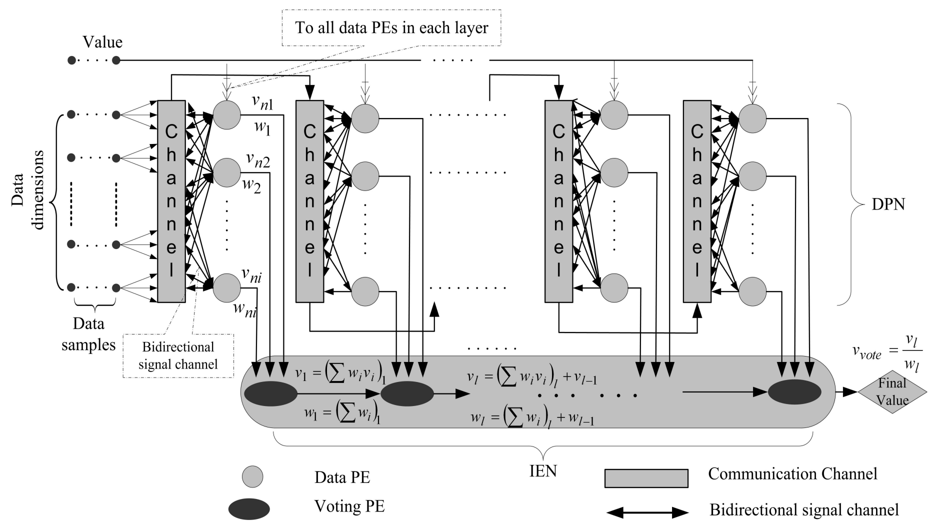

Fig. 2. Dynamic value system architecture

4 Simulation results

We illustrate the application of the proposed value system to the financial data analysis - bank prime loan rate prediction. Financial data prediction is difficult due to the inherent noise, non-stationary, and non-linear characteristics of such data sets. The neural network based approach is a powerful tool financial data analysis and many research results have been reported recently. For instance, in [6], three neural network based learning mechanisms, including standard back-propagation (SBP), scaled conjugate gradient (SCG) and backpropagation with Bayesian regularization (BPR) were used to predict the foreign currency exchange rates. In [7], foreign exchange rate prediction was analyzed using recurrent neural networks and grammatical inference. In [8], an iterative evolutionary learning algorithm using fuzzy logic, neural networks, and genetic algorithm was proposed for the financial data prediction, and prediction results were compared with those obtained by classical fuzzy neural networks as in [9]. In [10], J. Yao and C. Tan presented empirical evidence that a neural network model is capable of foreign exchange rates prediction, and they also discussed the network architecture, model parameters, and performance evaluation methods.

In this paper, we have used the dataset from Financial Forecast Center (www.forecasts.org) and compared our prediction results with those of [8] [9]. The feature vector has four dimensions (monthly bank prime loan rate, discount rate, federal funds rate and ten-year treasury constant maturity rate) and the prediction value is the next month's bank prime loan rate. We use the data set from January 1995 to December 2000 for training, and February 2001 to September 2002 for testing. Fig 4 shows the testing performance of the bank prime loan rate.

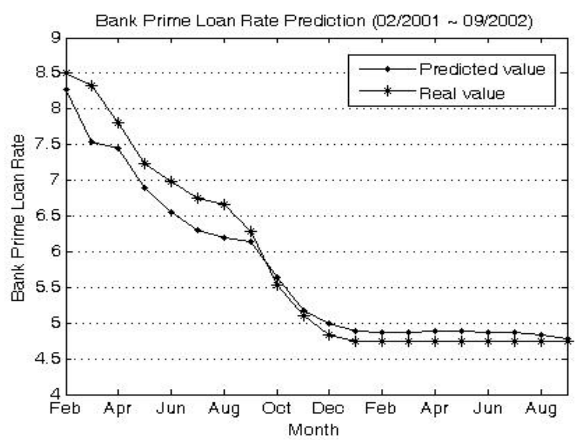

Fig. 3. Bank prime loan rate prediction by value system from February 2001 to September 2002

Fig 4 shows the mean square error (MSE) comparison of the proposed value system with the best results of hybrid evolutionary fuzzy neural network and genetic fuzzy neural learning algorithm (both of them with 300 training iterations) as presented in [8] and [9], respectively. As we can see from Fig. 3 and Fig. 4, the proposed value system can effectively learn and predict the signal values.

5 Conclusion

A novel online value system for machine learning is proposed. The proposed system combines a data processing network and information evaluation network. A dynamic three-curve fitting scheme is proposed to improve the fitting quality based on the statistical distribution of the data samples. In addition, a hardware-oriented system level architecture with hybrid shift-register channel structure is

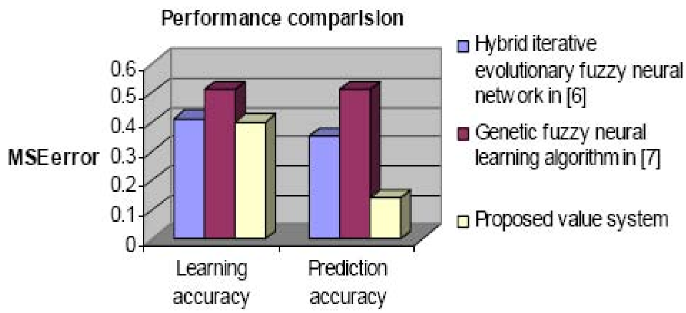

Fig. 4. Performance comparison of the proposed value system with those of [8] and [9]

also presented. Simulation results on the financial data prediction illustrate the effectiveness of the proposed value system. Motivated by the results presented in this paper, we believe that this approach may benefit research of value based reinforcement learning schemes.

References

1. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction. MIT Press, Cambridge, MA (1998)
2. Panayiotou, P.A., Pattichis, C., Jenkins , D., Plimmer, F.: A modular artificial neural network valuation system. The 10th Mediterranean Electrotechnical Conference (MELECON). (2000) 457 - 460
3. Babyey, S., Momtahan, O., Meybodi, M.R.: Multi mobile robot navigation using distributed value function reinforcement learning. Proceedings of IEEE International Conference on Robotics and Automation (ICRA '03), (2003) 14-19
4. Starzyk J. A., Zhu Z., Liu T.-H.: Self-Organizing Learning Array. IEEE Trans. on Neural Networks, vol. 16, no. 2, (2005) 355-363
5. Hecht-Nielsen, R.: Theory of the backpropagation neural network Int. Joint Conf. Neural Networks (IJCNN), vol.1, (1989) 593 - 605
6. Kamruzzaman, J.: ANN-based forecasting of foreign currency exchange rates, Neural Information Processing. 3(2), (2004) 49-58
7. Giles, C., Lawrence, S., Tsoi, A.: Noisy time series prediction using recurrent neural networks and grammatical inference. Machine Learning, vol. 44, (2001) 161-183
8. Yu, L., Zhang, Y: Evolutionary fuzzy neural networks for hybrid financial prediction. IEEE Trans. on Systems, Man, and Cybernetics, part C, (2005) 1-5
9. Zhang, Y., Kandel, A.: Compensatory genetic fuzzy neural networks and their applications. Series in Machine Perception and Artificial Intelligence, Singapore: World Scientific (1998)
10. Yao, J., Tan, C. L.: A case study on using neural networks to perform technical forecasting of forex. Neurocomputing, 34 (2000) 79-98